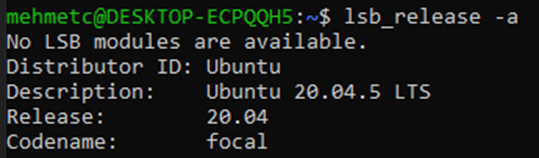
**CASEWORK OF ETERATION**

1. **SETUP OF THE WORK SYSTEM**

**1.1 Installing Ubuntu**

I installed Ubuntu 20.04 from Microsoft Store instead of Virtual Machine because it was more efficient for short tasks than Virtual Machine. After installing Ubuntu 20.04, I checked the version for a requirement which allows the use of ROS Noetic.



**1.2 Installing ROS Packages**

I wrote commands to install ROS Packages. [1]

|  |
| --- |
| ***sudo apt install ros-noetic-desktop-full***  ***sudo apt install python-rosdep*** |

**1.3 Creating Catkin Workspace**

I wrote commands to create Catkin Workspace. [2]

|  |
| --- |
| ***mkdir catkin\_ws*** *// creating a catkin package.*  ***cd catkin\_ws*** *//change directory for the catkin package.*  ***mkdir src*** *//creating source page.*  ***catkin\_make*** *//building catkin.*  ***catkin\_create\_pkg composiv\_tryouts std\_msgs rospy roscpp*** *//creating standard messages library with composiv\_tryouts, adding rospy and roscpp packages.* |

**1.4 Creating the Talker and Listener Nodes**

I created two sub-files, *composiv\_talker.py* and *composiv\_listener.py,* below the *composiv\_tryout* file.

1. **IMPLEMENTATIONS**

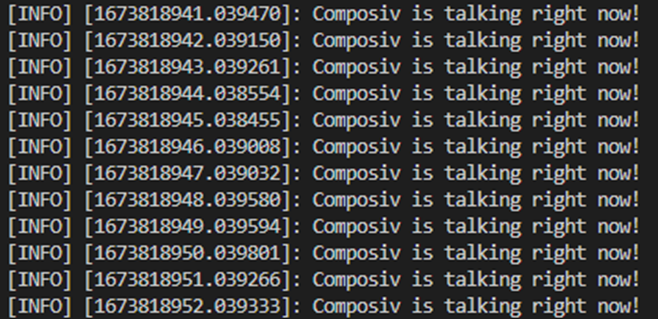
**2.1 Implementation of the Publisher (Talker)**

Firstly, I executed the *roscore* command in the new terminal.

I wrote example code into the file *composiv\_talker.py,* which was published in an internet source. [3]

I executed the *python composiv\_talker.py* commandin the terminal and I saw that it works.

I defined required libraries in this code, as well as publisher messages, publishing frequency, and size. and finally,I provided the message is displayed at the terminal.



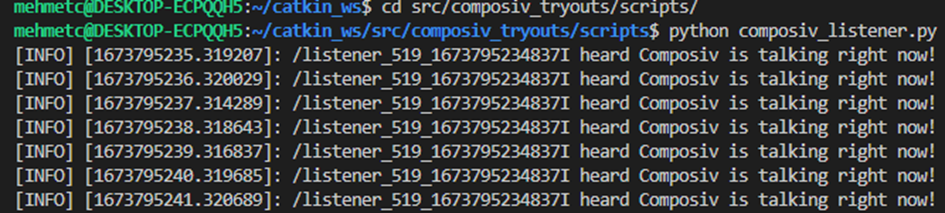
**2.2 Implementation of the Subscriber (Listener)**

Firstly, I executed the *roscore* command in the new terminal.

I wrote example code into the file *composiv\_listener.py,* which was published in an internet source. [3]

I executed the *python composiv\_talker.py* commandin the terminal. After that, I executed the *python composiv\_listener.py* commandin another terminal.

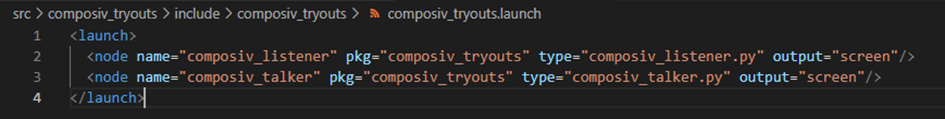
I defined required libraries in this code, as well as subscriber messages and I provided the messages are displayed at the terminal.

****

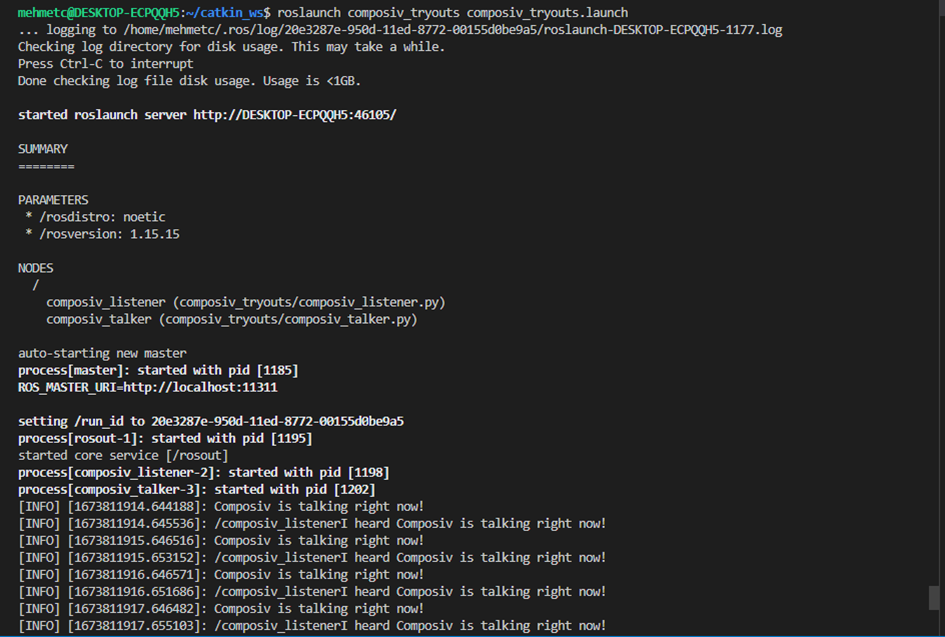
**2.3 Creating and Implementation of the Launch File**

I created the *composiv\_tryouts.launch* file in order for the publisher and subscriber to work together.

In this file*,* I defined node names, packages, and types of files.[4]



I executed the *roscore* command in the terminal. Then, I executed the *source devel/setup.bash* command in another terminal.[5] After that, I executed *roslaunch composiv\_tryouts composiv\_tryouts.launch* command.



1. **UNIT TEST**

I created the *test\_talker.py* file for the unit tests of publisher and subscriber.

Firstly, I executed the *roscore* command in the new terminal.

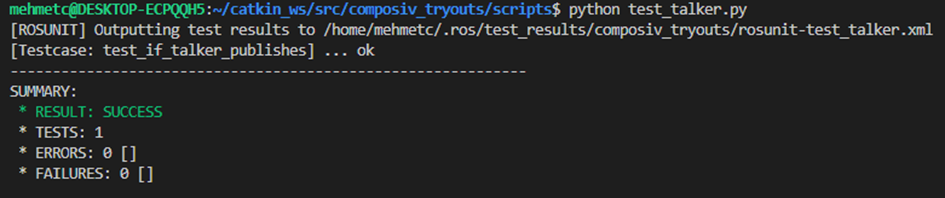
I wrote example code into the file *test\_talker.py,* which was published in an internet source.[6]

In this code, I defined test libraries. Then, I controlled publisher and subscriber communication between them.

I execute the *roslaunch composiv\_tryouts composiv\_tryouts.launch* command at another terminal.

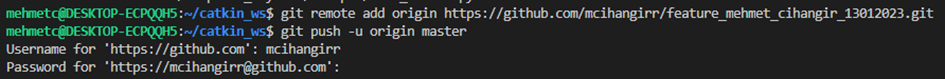
After that, I executed the *python test\_talker.py* command at another terminal.

I noticed that the test went well.

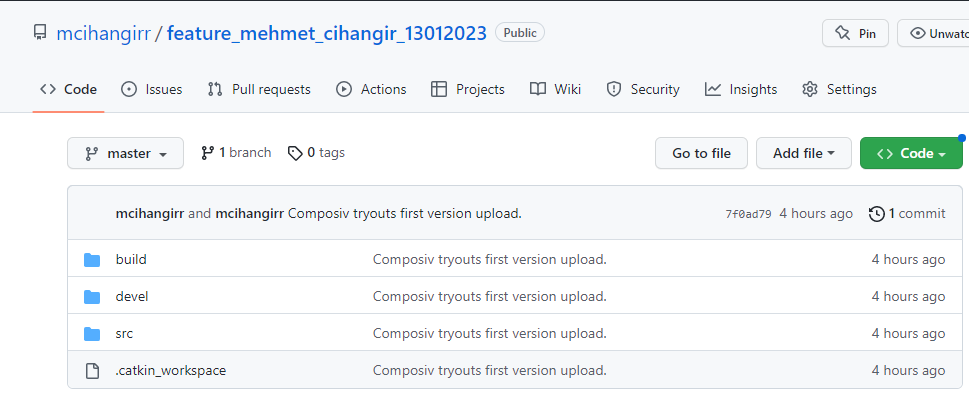


1. **CREATING A REPOSITORY AND UPLOADING TO GITHUB**

I wrote these commands to push my project to github.

****

Firstly, I created a repository named “features\_mehmet\_cihangir\_13012023”. Then, I uploaded my project file below the Master Branch.



1. **REFERENCES**

[1] <http://wiki.ros.org/Installation/Ubuntu>

[2] <http://wiki.ros.org/catkin/commands/catkin_make>

[3] <http://wiki.ros.org/rospy_tutorials/Tutorials/WritingPublisherSubscriber>

[4] <https://automaticaddison.com/how-to-create-and-execute-ros-launch-files/>]

[5] <https://get-help.robotigniteacademy.com/t/rl-exception-error-neither-a-launch-file-package-nor-launch-file-name/20325/4>

[6] <https://www.theconstructsim.com/ros-qa-098-see-ros-publisher-works-using-ros-unit-testing/>